EXHIBIT D

Library of Parameterized Hardware Modules for Floating-Point Arithmetic with An Example Application

A Thesis Presented

by

Pavle Belanović

to

Department of Electrical and Computer Engineering

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of

Electrical Engineering

Northeastern University

Boston, Massachusetts

May 2002

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Library of Parameterized Hardware Modules for Floating-Point Arithmetic with An Example Application.

Author: Pavle Belanović.

Department: Electrical and Computer Engineering.

Approved for Thesis Requirements of the Master of Science Degree

Thesis Advisor: Miriam Leeser	Date
Thesis Reader: Dana Brooks	Date
Thesis Reader: Waleed Meleis	Date
Chairman of Department: Fabrizio Lombardi	Date

Graduate School Notified of Acceptance:

Director of the Graduate School	Date

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Library of Parameterized Hardware Modules for Floating-Point Arithmetic with An Example Application.

Author: Pavle Belanović.

Department: Electrical and Computer Engineering.

Approved for Thesis Requirements of the Master of Science Degree

Thesis Advisor: Miriam Leeser	Date
Thesis Reader: Dana Brooks	Date
Thesis Reader: Waleed Meleis	Date
Chairman of Department: Fabrizio Lombardi	Date

Graduate School Notified of Acceptance:

Director of the Graduate School	Date

Copy Deposited in Library:

Reference Librarian	Date

The simulator used to test the VHDL descriptions was Mentor Graphics ModelSim (see Section 1.3). Iteration between modification of the VHDL description and analysis in the simulator continued until the correct operation of the module was achieved for all the test vectors.

The second testing stage was done in hardware. The VHDL description, shown to operate correctly in the simulator, was synthesized and loaded on the Wildstar board. The same set of test vectors used in simulation was applied to the hardware implementation. Testing results from hardware were compared to expected result values. If they did not match, the VHDL description was further modified to achieve the correct operation.

An example of a test vector used to test the IEEE single precision adder circuit in Section 2.5 is given below.

Operand 1	**Operand 2**	**Sum**
41BA3C57	4349C776	43610F01

$$41BA3C57_{16} =$$

$$= 0\ 10000011\ 01110100011110001010111_2$$

$$= +\ 1.45496642 \times 2^4$$

$$= 23.27946281$$

$$4349C776_{16} =$$

$$= 0\ 10000110\ 10010011100011101110110_2$$

$$= +\ 1.57639956 \times 2^7$$

$$= 201.77914429$$

$$
\begin{aligned}
Sum = & \\
& = 225.05860710 \\
& = +\,1.75827036 \times 2^{7} \\
& = 0\ 10000110\ 11000010000111100000001_2 \\
& = 43610F01_{16}
\end{aligned}
$$

2.7 Results

This section presents results of synthesis experiments conducted on the floating-point operator modules *fp_add*, *fp_sub* and *fp_mul* of Section 2.3. The aims of the experiments are to:

- determine the area of the above modules in several floating-point formats,
- examine the relationship between the area and total bitwidth of the format, and
- estimate the number of modules that can realistically be used on a single FPGA.

The experiments were conducted by synthesizing the modules for specific floating-point formats on the Annapolis Micro Systems Wildstar reconfigurable computing engine (see Section 1.3). Table 2.2 shows results of the synthesis experiments on floating-point operator modules. The quantities for the area of each instance are expressed in slices of the Xilinx XCV1000 FPGA. Results for the `fp_add` module in Table 2.2 also represent the `fp_sub` module, which has the same amount of logic.

Floating-point formats used in the experiments were chosen to represent the range of realistic floating-point formats from 8 to 32 bits in total bitwidth and include the IEEE single precision format (E1 in Table 2.2).

Table 2.2: Operator synthesis results

Format	**Bitwidth**			**Area**		**Per IC**	
	total	exponent	fraction	fp_add	fp_mul	fp_add	fp_mul
A0	8	2	5	39	46	236	200
A1	8	3	4	39	51	236	180
A2	8	4	3	32	36	288	256
B0	12	3	8	84	127	109	72
B1	12	4	7	80	140	115	65
B2	12	5	6	81	108	113	85
C0	16	4	11	121	208	76	44
C1	16	5	10	141	178	65	51
C2	16	6	9	113	150	81	61
D0	24	6	17	221	421	41	21
D1	24	8	15	216	431	42	21
D2	24	10	13	217	275	42	33
E0	32	5	26	328	766	28	12
E1	32	8	23	291	674	31	13
E2	32	11	20	284	536	32	17

The number of operator cores per processing element, shown in the two rightmost columns, is based on a Xilinx XCV1000 FPGA, with a total of 12288 slices, with 85% area utilization. A realistic design cannot utilize all the resources on the FPGA because of routing overhead; a practical maximum is estimated at about 85%. Also included is an overhead allowance of approximately 1200 slices for necessary circuitry other than the operators themselves. This allowance may represent memory read and/or write circuitry, state machines, register tables and similar circuits which are required by most designs. Thus, results shown for the number of modules per IC correspond to realistic designs using the operator cores.

The results in Table 2.2 show growth in area with increasing total bitwidth, for all modules. This growth is represented graphically in Figure 2.10.